
Introducción a Ruby on Rails

— Como crear y correr una
aplicación —

Overview

- Cómo crear una aplicación
- Cómo ejecutar una aplicación
- Estructura de directorios
- Agregar páginas estáticas a la aplicación

Creando una aplicación

- Ir hasta el directorio donde se quiere crear la aplicación
- **rails new my_first_app**
- Para tener más opciones:
rails new -h

```
→ modulo-3 git:(master) X rails new my_first_app
create
create  README.rdoc
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/assets/javascripts/application.js
create  app/assets/stylesheets/application.css
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/views/layouts/application.html.erb
create  app/assets/images/.keep
create  app/mailers/.keep
create  app/models/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/bundle
create  bin/rails
create  bin/rake
create  config
```

Bundler (gems manager)

- Al final de la creación, la última línea dice:
 - **run bundle install**
- Es parte de la creación de una aplicación rails, obtiene las dependencias.

```
→ my_first_app git:(master) X bundle install

Fetching gem metadata from https://rubygems.org/.....
Fetching version metadata from https://rubygems.org/...
Fetching dependency metadata from https://rubygems.org/..
Resolving dependencies...
Rubygems 2.0.14 is not threadsafe, so your gems will be insta
l gem installation.
Installing rake 11.3.0
Using i18n 0.7.0
Using minitest 4.7.5
Installing multi_json 1.12.1
Using thread_safe 0.3.5
Installing tzinfo 0.3.52
Using builder 3.1.4
Using erubis 2.7.0
Using rack 1.5.5
Using mime-types 1.25.1
Using polyglot 0.3.5
Using activerecord-deprecated_finders 1.0.4
Using arel 4.0.2
Using bundler 1.13.5
Installing coffee-script-source 1.10.0
Using coffee-script 2.7.0
```

Control de versiones en una aplicación Rails

- Rails automáticamente genera un archivo **.gitignore**
- **cd my_first_app**
- **git init**
- **git add .**
- **git commit -m "Initial commit"**



El repositorio Git
debe estar dentro!

Ejecutando la aplicación

- Ahora tenemos el directorio **my_first_app** con la estructura autogenerada / algo de código también.
- Rails cuenta con un servidor embebido.
- Ejecución:
- Abrir una terminal y ubicarse en el **my_first_app**
 - **rails server** (o **rails s**)

Ejecutando la aplicación


VERSIÓN

ENVIRONMENT

DONDE?

```
→ my_first_app git:(master) x rails s
=> Booting WEBrick
=> Rails 4.0.4 application starting in development on http://0.0.0.0:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2016-10-30 17:44:21] INFO  WEBrick 1.3.1
[2016-10-30 17:44:21] INFO  ruby 2.0.0 (2014-05-08) [x86_64-linux]
[2016-10-30 17:44:21] INFO  WEBrick::HTTPServer#start: pid=13867 port=3000
```

Rails te muestra los pasos



Welcome aboard

You're riding Ruby on Rails!

[About your application's environment](#)

Getting started

Here's how to get rolling:

1. Use `rails generate` to create your models and controllers

To see all available options, run it without parameters.
2. Set up a root route to replace this page

You're seeing this page because you're running in development mode and you haven't set a root route yet.

Routes are set up in `config/routes.rb`.
3. Configure your database

If you're not using SQLite (the default), edit `config/database.yml` with your username and password.

Browse the documentation

[Rails Guides](#)
[Rails API](#)
[Ruby core](#)
[Ruby standard library](#)

Directorio de la aplicación

▼ my_first_app

▶ app

Controllers, views, models (and helpers)

▶ bin

▶ config

Archivos de configuración (Qué BD ?)

▶ db

Scripts de migración relacionados con la bd

▶ lib

▶ log

▶ public

Archivos estáticos

▶ test

▶ tmp

▶ vendor

.gitignore

Gemfile

Dependencias manejadas por el bundler

Gemfile.lock

README.rdoc

Rakefile

config.ru

.gitignore

LICENSE

README.md

public/hello_static.html

- Los archivos **estáticos** están en el directorio **public**
- Los servidores miran el directorio **public** antes de mirar cualquier otra cosa.
 - Si el archivo puede ser encontrado en **public**, simplemente se muestra.
- Entonces, si quisiéramos mostrar una página completamente estática, podríamos agregarlo a nuestro directorio **public**.

hello_static.html

```
hello_static.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Totally Static </title>
5   </head>
6
7   <body>
8     <h2> Will never generate dynamic content:( </h2>
9   </body>
10 </html>
```

< > ↺ 📄 localhost:3000/hello_static.html

Will never generate dynamic content:(

Entonces...

- **rails new** crea una aplicación nueva
- **rails s** (o **rails server**) ejecuta la aplicación
- Las páginas estáticas están en el directorio **public**.